Modification of Ant Colony Optimization Algorithm to Solve the Traveling Salesman Problem

Alda Larasati Anindya[1*)], Ketut Bayu Yogha Bintoro[2], Silvester Dian Handy Permana[3]
[1]Program Studi Teknik Informatika, Fakultas Industri Kreatif dan Telematika, Universitas Trilogi
[2]Program Studi Teknik Informatika, Fakultas Industri Kreatif dan Telematika, Universitas Trilogi
[3]Program Studi Teknik Informatika, Fakultas Industri Kreatif dan Telematika, Universitas Trilogi
email: [1]aldaanindya@trilogi.ac.id, [2]ketutbayu@trilogi.ac.id, [3]handy@trilogi.ac.id

Abstract **– Traveling salesman problem (TSP) is an optimization problem in determining the optimal route of a number of nodes that will only be passed once with the initial node as the final destination. One method for solving TSP is the Ant Colony Optimization (ACO) Algorithm. ACO is inspired by ant behaviour in searching for food, where ants produce pheromones to find food sources and make a route from the colony to food that will be followed by other ants. However ACO has not been considered as the optimal method for resolving TSP. This is because ACO has several shortcomings in the computational process. Comparisons between pheromones are not yet clear, and slow computing time causes the results of ACO to be not optimal. To correct these deficiencies, modifications will be made to the ACO. Modifications are made by changing some values in the ACO, such as adjusting the number of ants by the node automatically, changing the value in the pheromone renewal, and adding value to the construction of the solution. The outcome of this research is the modification of ACO did not provide shorter computing time with a more accurate final value, thus did not provide an optimal solution. The test results in this study found that the average computation time for the last iteration of each test was 0.54 second, and for the 10 iteration computation time obtained an average of 5.54 second for four tests. The amount of memory used in four tests in this study was 440.11 mb for 10 iterations.**

Keywords – Ant Colony Optimization, ACO Modification, Traveling Salesman Problem.

I. INTRODUCTION

Traveling Salesman Problem (TSP) is one of the classic problems of optimization that can be found in everyday life. TSP is stated as an optimization problem to find the most optimal route where a number of nodes will be passed and the node may only be passed once, with the initial node as the final destination. Some examples of TSP cases in daily life include the search for multiple locations, delivery of goods, and others. There are several methods developed to complete the TSP, one of which is Ant Colony Optimization (ACO) [4].

The Ant Colony Optimization (ACO) algorithm is inspired by ant behavior in foraging in nature. Based on a book written by Marco Dorigo and Thomas Stutzle in 2004 entitled "Ant Colony Optimization", ACO can be applied directly to the TSP case easily. This is because the behavior of ants when foraging has the same mechanism as the TSP case. Initially, ants will walk randomly when they first come out of their nest to find food, and leave pheromone trails when returning to their nests. Each ant has pheromones, and then when other ants find food, then the possibility of ants running randomly is getting thinner because ants tend to follow in the footsteps of pheromones left by the ants before. When that happens, the pheromone trail gets stronger and the ants will stop walking randomly until the food runs out. But pheromones can also experience evaporation, the longer the route, the faster the pheromone evaporation. The shorter path is more often traveled by ants, so that the intensity of the pheromone is more concentrated, and the evaporation is slower. Overall, it will be with high pheromone intensity. This makes ACO work well in the case of the TSP. [1]

However, ACO has several shortcomings so it is considered not to really provide optimal results to complete the TSP. The most prominent drawback is that the algorithm's computing time tends to be slow. This is because the pheromone comparison of each iteration does not have a clear comparison. In addition, the large number of ants also greatly affects the computational time, and tends to give suboptimal results. ACO also has a stagnation problem, where all ants are only concentrated in one pathway with the most pheromones, so the probability of finding another more optimal path becomes smaller.

Based on this brief description, modifications will be made to the ACO. Modifications are made by correcting the deficiencies that have been described without changing the base workings of the algorithm. The implementation of ACO modification is done by making a comparative simulation application between conventional ACO and modified ACO, this application uses the Java Language by using the Netbeans IDE. With this simulation application, the final results of the ACO modification can be compared with the Conventional ACO as a whole starting from the route formed until the most optimal computing time. With this modification, it is hoped that ACO can provide more optimal results, which can later be widely implemented, as well as a reference for further research.

II. LITERATURE REVIEW

In a study titled Ant Colony for the TSP conducted by Yinda Feng in 2010, the results provided by ACO have many advantages, but also many disadvantages. The drawback of the results of this study is that ACO requires a slow computation time. This happens because the pheromones produced do not have a significant difference from each iteration. In this study it was also explained that ACO has a stagnation problem, where all ants are only concentrated on one pathway with the most pheromones.

The following year, in a study titled Solving The Traveling Salesman Problem Using The Ant Colony Optimization conducted by Ivan Brezina Jr. and Zuzana Cickova concluded that the quality of the final product depends on the number of ants. Fewer ants affect route changes. This causes the ACO algorithm to only produce optimal results with a small number of ants, because the fewer ants, the fewer iterations are carried out. Meanwhile, to solve the problem with a large number of nodes, it takes a greater number of ants and iterations. This problem is similar to previous studies where the ACO computation time tends to be slow.

Dedy Mulia conducted a study in the same year titled Application of the Ant System Algorithm (AS) in the Case of the Traveling Salesman Problem (TSP) provided a detailed explanation for solving TSP using conventional ACO. ACO which is implemented into TSP is able to produce the shortest route length. In this study only explained how the ACO process is applied to a simple TSP case, so it is not known how optimal ACO is in solving TSP with a large number of nodes.

Nitun N. Poddar and Devinder Kaur in 2013 further researched in modifications to the ACO algorithm were carried out in their study titled Solving the Traveling Salesman Problem using Reinforced Ant Colony Optimization Techniques in 2013. In this study, modifications were made in the form of reinforcement, where changes were made to the original algorithm of ACO by changing several values. . This is done so that routes that were previously considered not optimal to pass, and give better results than before. Although the gain from this gain is better than the unmodified ACO in terms of tour distance, the convergence rate of this modified ACO is also higher. This causes the computation time to find another tour in the next iteration is also slower.

Yousefikhoshbakht et al. conducted a study in 2013 titled Modification of the Ant Colony Optimization for Solving the Multiple Traveling Salesman Problem, in which modifications were made in the form of changing several values in the equation in conventional ACO. Changes are made in the solution development stage, where the value of the combined distance between two nodes is added in the equation. The end result of this study is a shorter computation time and more stable memory usage. Although the results are better than conventional ACO, the final result does not provide a significant change.

Further research was conducted in 2016 by Abdulqader Mohsen titled Annealing Ant Colony Optimization with Mutation Operator for Solving TSP. In this study, a modification of the ACO was also carried out as in previous studies. Modifications include strengthening and hybridization of ACO and mutation operators. When compared with the unmodified ACO, the last iteration of this modification gives more optimal results without stagnation or premature convergence. However, this modification is still not fully optimal.

The final research that will be used as a benchmark in this study was conducted in 2018 by Xu et al. titled Dynamic Vehicle Routing Problems with Enhanced Ant Colony Optimization. This study made a modification in the form of ACO strengthening in the form of a merger between ACO and K-means to solve dynamic problems. The result of this modification has the advantage of finding the optimal solution. However, the biggest drawback of this modification is that the convergence time is slower than the ACO which has not been modified at a later stage.

III. RESEARCH METHOD

Based on the literature review, there are quite a bit of research gap on the Ant Colony Optimization method in general. Mainly when implementing the ACO to the Traveling Salesman Problem, ACO tends to be slow in the computation process. This is due to the increasing number of ants affecting memory usage, and insignificant changes in pheromones that do not give optimal results. In addition, even though there have been modifications in the form of changes in value and reinforcement by combining ACO with other optimization algorithms, the final results given did not make any significant changes.

Based on this analysis, the biggest drawback of this algorithm is in the comparison stage of each iteration. The updated pheromone does not necessarily compare with the pheromone value from the previous iteration. This has an effect on the final result of the solution, because to determine the optimal result requires comparisons that have a definite value in each iteration.

In addition, it is not explained if after all the iterations have been successfully executed, the resulting route is really optimal, because basically the agent (ant) generates pheromones randomly. The number of ants also affects computing performance. A large number of ants produces a more accurate final result, but slows down the computation time, whereas a smaller number of ants results in a faster computation time, but the final result is less accurate.

A. *Modified Ant Colony Optimization*

The ACO stage starts when the ants walk around in random when foraging. By the time the ants went back to the nest or starting point, the ants that left a pheromone trail for other ants to follow before it starts to evaporate. When other set of ants follow the path that contains previous pheromones, the pheromone's trail gets stronger and the ants will stop running randomly. Over time, the intensity of the pheromones will be more concentrated and an optimal path will be formed. In addition to pheromones intensity, other factors in determining the optimal path is the intensity of the distance between nodes.

Figure 1 shows the proposed method of modification of the ACO algorithm to fit the needs in solving the

Traveling Salesman Problem, based on the brief description above

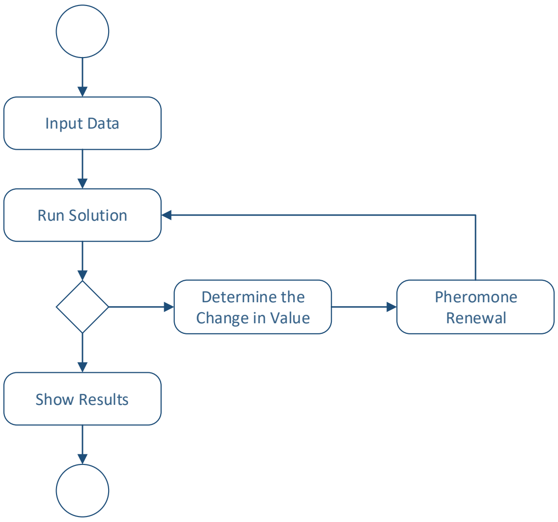

Figure. 1 The Proposed Method for Modifying the ACO

1. *Automatic Ant Deployment*

This modification is done to fix the problems of the ACO where a large number of ants causes stagnation. Therefore, this modification is made so the number of ants automatically adjust to the number of nodes. The conditions can be seen in the formula as follows.

$$\begin{aligned}&if\ node \geq 50,\ then\ k = 50\\&if\ node > 50,\ then\ k = 0{,}25 * node\end{aligned} \tag{1}$$

In addition to further decrease the computation time, this modification were made to improve the quality of the solution.

2. *Pheromone Intensity Renewal*

The modifications in this stage are made by adding some values to the pheromone renewal formula. Here are the formula for pheromone intensity renewal.

$$\begin{aligned}&\tau_{ij}(t+1) \leftarrow (1-p)\tau_{ij}(t) + \Delta\tau_{ij}(t) + b \cdot \tau^{b}{}_{ij}\\&where\\&\Delta\tau_{ij}(t) = \Sigma\tau_{ij}^{k}, k = 1, \dots N,\\&\Delta\tau_{ij}^{k}(t) = \frac{1}{L^{k}(t)}\ \ if\ \ (i,j) \in T^{k}(t)\ \ and\ 0\\&otherwise\\&\Delta\tau_{ij}^{b}(t) = \frac{1}{L_{best}(t)}\ \ if\ \ (i,j) \in\\&best\ path\ and\ 0\ otherwise\end{aligned} \tag{2}$$

3. *Modify Main Solutions*

Modification are made to the main ACO solution by adding an extra parameter to the solution, which is calculated based on the following formula.

$$P_{ij}^{k} = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [n_{ij}]^{\beta} \cdot [\kappa_{ij}]^{\lambda}}{\Sigma[\tau_{ij}(t)]^{\alpha} \cdot [n_{ij}]^{\beta} \cdot [\kappa_{ij}]^{\lambda}} \quad if\ \ j \in J_{ik} \tag{3}$$

The parameter κ_{ij} a distance combination between two nodes, while the λ is the controller parameter. κ_{ij} is calculated by the following formula.

$$\kappa_{ij} = \alpha \cdot d_{ij} + \beta \cdot d_{ij} - \lambda \cdot d_{ij} \tag{4}$$

B. *Testing and Comparison*

Comparisons were made to measure the final results of conventional ACO and modified ACO. After the final results are obtained, the next step is verification if modified ACO provides better and optimal results when compared to conventional ACO. Modified ACO gives better and optimal results compared to conventional ACO if all of these criteria have been met.

1. The resulting route in modified ACO has a shorter total distance
2. Faster modified ACO computing time
3. Memory that is used for computing is relatively lighter

The criteria above are determined based on the estimation of the stage of the research method, where the modification of the number of automatic ants and additional variables at the solution development stage should provide a better change in terms of computational time and the final route produced is more optimal. If one of the criteria is not met at the time of testing, the modified ACO does not provide optimal results.

C. *Parameters of Research*

For this research, several parameters are required in order to determine the limitations. The first parameters to determine is the control parameter. These parameters serve to control the main solution phase in the modified ACO, where the values will not change no matter the pheromone intensity. There are three different parameters as shown in Table 1 below.

Table 1. Control Parameters in Main Solution Phase

Alpha (α)	0.5
Beta (β)	0.7
Lambda (λ)	0.6

As shown in Table 1, Alpha (α) is a parameter that controls the initialization value of pheromone in each iteration, while Beta (β) is a parameter that controls the value of the visibility of the distance. Furthermore, the Lambda value (λ) is a parameter controlling the value of the combined distance between nodes.

Aside from the control parameters used in the main solution, there needs to be limitations on other parameters. These parameters are determined by the user, and there are three different parameters as sown in Table 2.

Table 2. User determined parameter limits

Ant (k)	5-50
Evaporation rate	0.01-0.99
Maximum Iteration	Determined by user

Based on Table 2, the minimum and maximum number of ant (k) affects the determination of the best route in terms of computation time. The pheromone evaporation rate correlates with the number of ant (k) in determining the best route determined by conventional ACO or Modified ACO. This is due to the estimation that the faster or slower the pheromone evaporation will determine the percentage of ant (k) to form a route from a node, especially in computation for the development of a Modified ACO solution. Maximum iteration parameter to determine how many iterations are executed when comparing Conventional ACO and Modified ACO, because if it is not given a limit, the computation will continue without stopping.

IV. RESULTS AND DISCUSSION

In testing and comparing the results of conventional ACO and modified ACO, several research samples are needed. These samples are needed to find out how optimal the results of modified ACO are when applied to several different cases. There are four data inputted to determine the results. These data are the number of ant (k), number of nodes, evaporation rate, and maximum iteration. If the data has been inputted, then the case is determined in combination with the following.

1. The number of ant (k) is 50% more than the number of nodes, with an evaporation rate of 1% of the number of nodes.
2. The number of ant (k) is 30% more than the number of nodes, with an evaporation rate of 4% of the number of nodes.
3. The number of ant (k) is 30% less than the number of nodes, with an evaporation rate of 1% of the number of nodes.
4. The number of ant (k) is 50% less than the number of nodes, with an evaporation rate of 4% of the number of nodes.

The provisions of the case for this study are based on estimates from the provisions in the previous chapter. The number of ant (k) affects the accuracy of the route determination in Conventional ACO computing, with the estimation that the number of ant (k) will give more accurate results than the number of ant (k) is small, but with more memory usage. The number of ant (k) does not affect the Modified ACO computational stage, where the number of ant (k) automatically follows how many number of nodes, if the number of nodes is 50 then the number of ant (k) is also 50, whereas if the number of nodes is above 50. Next is the explanation for the provisions of the pheromone evaporation rate, where the density of the evaporation rate influences in determining the route at the main solution stage because ant (k) will be more consistent in following the route if the pheromone density is high.

1. *Large Number of Ants with Low Evaporation Rate*

In this case, each variable will be entered with the condition that the value of ant (k) is greater than the number of nodes inputted, with a low evaporation rate. The value ant (k) is made of 50% larger than the number of nodes, and the evaporation rate is 1% of the number of nodes. This provision is based on an estimation where more ant (k) values give more accurate results when compared to a smaller number, this is because the greater number of ant (k) to compute between nodes gives more pheromones. However, this is not the case if the evaporation rate inputted is not concentrated, because the route formation information by ant (k) is influenced by the pheromone density.

The first variable to be determined is a node which amounts to 25 nodes. Furthermore, the specified variable is the number of ant (k), according to the provisions of the first case, the number of ant (k) is 50% more than the number of nodes, then the value is 37 ant. The last variable specified is the evaporation rate, based on the condition that the value is 1% of the number of nodes, then the value is 0.25. The test results may be seen on figure 2 down below.

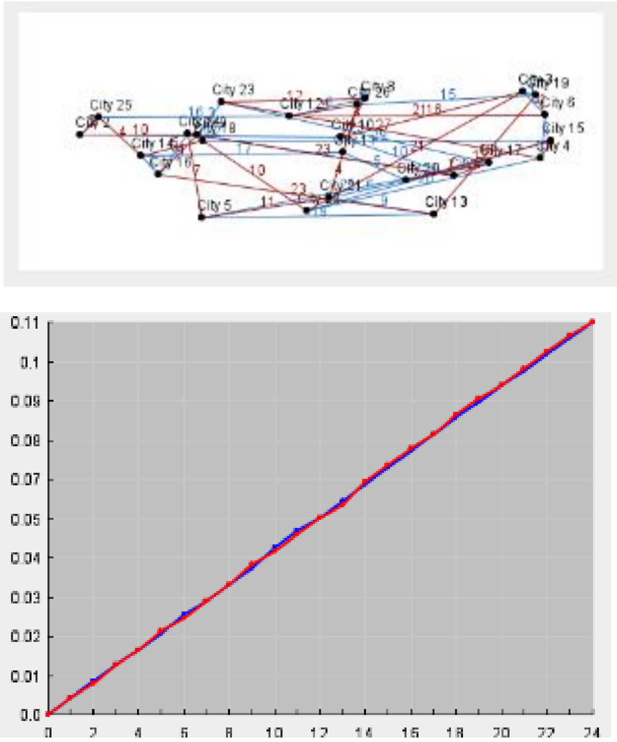

Figure. 2 Test Results for the first case

According to the test result, there are a number of differences between the conventional ACO and modified ACO. Specifically, the shape of the routes is very different from each other, and from a visual perspective it appears that the route formed by modified ACO is more random when compared to conventional ACO. Even so, in terms of computing time, modified ACO is more stable than conventional ACO. The time elapsed for this test is 0.257 second for the last iteration, and 2.761 second for a total of 10 iteration. The memory usage for this test is 29.15 mb for the last iteration, and 321.25 mb for a total of 10 iteration.

2. *Large Number of Ants with High Evaporation Rate*

In the second case that has been previously determined, the variables are in accordance with the provisions where the value of ant (k) is greater than the number of nodes inputted, with a thick evaporation rate, different from the previous case. For the ant value variable (k) is 30% more than the number of nodes, then for the evaporation rate variable value 4% of the number of nodes. As in the previous case, this provision is based on an estimate where the value of ant (k) which is more than the number of nodes gives more accurate results when compared to the number that is less, the difference is in the number of ant (k) specified. The process of forming one of the routes is influenced by the level of pheromone density, the determination of the concentrated evaporation rate is estimated to give more optimal results, especially if combined with the large number of ant (k).

The first variable to be determined is the number of nodes, which is 20 nodes. The next specified variable is the number of ant (k), provided that the number of ant (k) is 30% more than the number of nodes, so the number of ant (k) for this test is 26 ant. For the last variable determined is the evaporation rate. The second case condition is a thick evaporation rate, which is 4% of the number of nodes, therefore the evaporation rate determined for this test is 0.8. After all variables have been determined, the test results can be seen in figure 3 below.

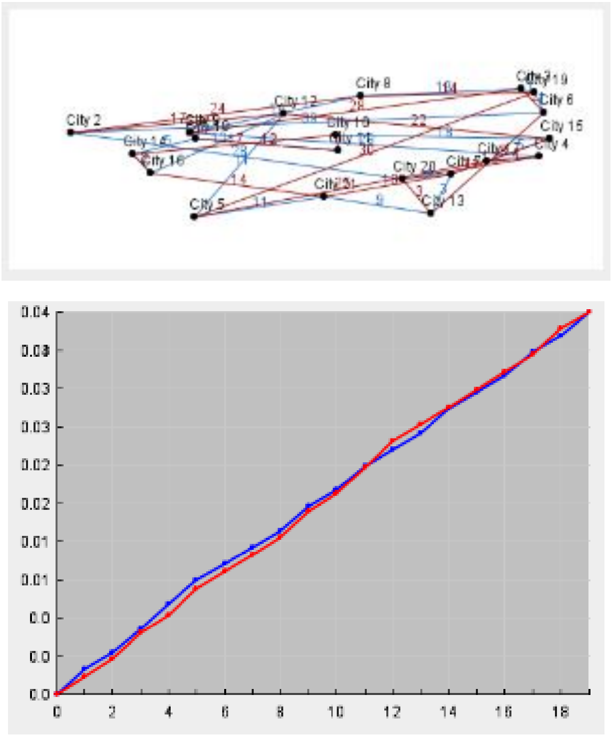

Figure. 3 Test Results for the second case

The final results of the second case are the computational time and the route formation of conventional ACO and modified ACO experienced some difference, but not significantly. This is caused by the thick evaporation rate, which causes the distribution of ant (k) to be more evenly distributed so that the formation of the route is tidier when compared to the first case where the evaporation rate is not as concentrated. As with the first case, the routes generated in the last iteration of conventional ACO and modified ACO are just as random, but the routes generated by modified ACO tend to be more random because many route formations overlap one another. This is caused by the number of ant (k) determined in modified ACO is greater than ant (k) in conventional ACO. The time elapsed for this test is 0.108 second for the last iteration, and 1.149 second for a total of 10 iteration. The memory usage for this test is 26.72 mb for the last iteration, and 279.21 mb for a total of 10 iteration.

3. *Small Number of Ants with Low Evaporation Rate*

For the third of the four predetermined cases, there are three variables to be determined. The first variable is the value of ant (k) is smaller than the number of nodes determined, with a non-concentrated evaporation rate. The ant (k) variable determined for this case is 30% less than the number of nodes, then for the evaporation rate variable value is 1% of the number of nodes. Determination of the variables for this case is based on estimation if the value of ant (k) which is less than the number of nodes gives more accurate results with smaller memory usage when combined with the same small evaporation rate variable.

For this case, several variables are determined to carry out this test. The first variable to be determined is nodes, the amount of which is 50 nodes. The next variable to be is determined is ant (k), according to the provisions stated, the number of ant (k) is 30% less than the number of nodes, so the number is 35 ant. The last variable to be determined is the evaporation rate, with a provision being 1% of the number of nodes, the evaporation rate is 0.5. Once the variables are determined, the results can be seen in figure 4 below.

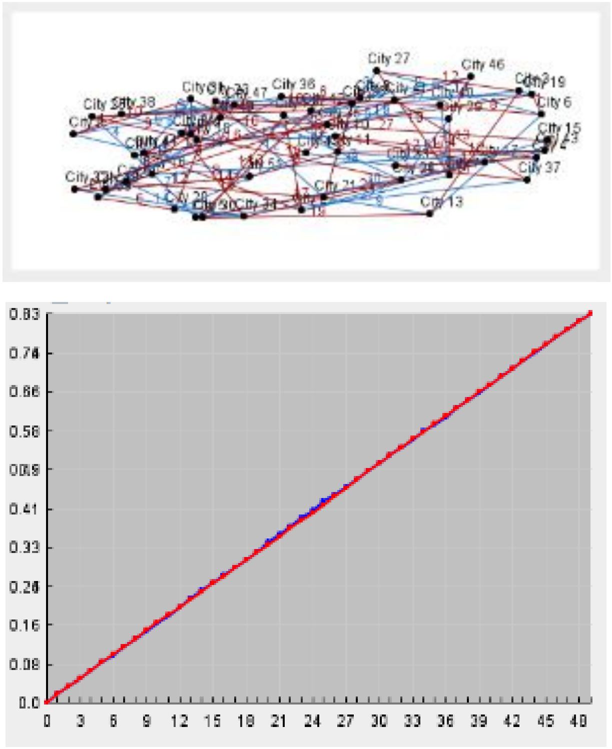

Figure. 4 Test Results for the third case

In this case, variables are large enough that the results give a significant difference. The final result of this test is that the routes generated from conventional ACO and modified ACO have some differences, where the routes produced by conventional ACO are more evenly distributed, and the routes produced by modified ACO are more random and overlapping even though it is not the case. The computing time of a conventional ACO is slower than that of a modified ACO. In this test the time elapsed at the last iteration is 1,706 seconds, and at 10 iterations is 17,350 seconds. Memory usage for this test is 65.51 mb for the last iteration, and 804 mb for 10 iterations.

4. *Small Number of Ants with High Evaporation Rate*

For the last case, the variables are determined to achieve a different outcome than the previous three cases. The variables determined for this case are ant (k) values smaller than the number of nodes determined, combined with a thick evaporation rate. The ant value variable (k) determined for this case is 50% less than the number of nodes, then for the evaporation rate variable value is 4% of the number of nodes. The variable provisions for this case are based on estimation if a smaller ant (k) value compared to the number of nodes gives a smaller memory usage, and when combined with a concentrated evaporation rate variable, will give an optimal final result.

As with the previous cases, several variables are determined to carry out this test. The first variable that is determined is the node, which in this study was determined as many as 22 nodes. Next, the ant (k) variable is determined, the value of ant (k) is 50% less than the number of nodes, so the total is 11 ant. The last variable specified is the evaporation rate, in this case the value is determined to be 4% of the number of nodes, and therefore the evaporation rate is 0.88. After these variables have been determined, the result of this test can be seen in figure 5 as follows.

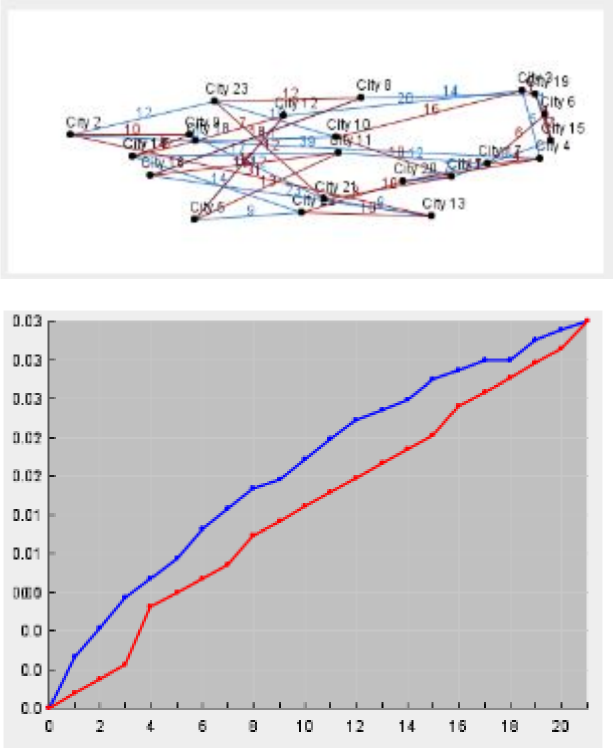

Figure. 5 Test Results for the fourth case

The route formed by conventional ACO is more neat and structured compared to modified ACO. Visually, the route formed by conventional ACO is spread evenly between nodes. One of the influencing factors is the number of ant (k) whose numbers are much less than the ant (k) in Modified ACO which is 50 ant (the provisions based on equation 3.6 in the previous chapter). In Modified ACO, the routes that are formed are more random and overlapping, so that visually seen many nodes are traversed many times even though what happened is not the case. Overall, both conventional ACO and modified ACO in this test have their advantages and disadvantages, but they are not consistent in terms of the route and computational time produced. Where conventional ACO provides better route formation results, the computation time is slower compared to modified ACO. The opposite occurs in modified ACO, where the computational time produced is more optimal and stable, the resulting route is more random and overlapping. In this test the time elapsed at the last iteration is 0,089 seconds, and at 10 iterations is 0,900 seconds. Memory usage for this test is 27.69 mb for the last iteration, and 355.99 mb for 10 iterations.

V. CONCLUSION

Based on the result of this study, it is concluded that Modified ACO does not provide better results than conventional ACO in terms of computational speed and the final route that is formed. From several tests conducted, the final route produced by Modified ACO results is not consistent, where in some samples it looks more optimal but in other samples it is even more random and overlapping. Modified ACO affects memory usage when computing for several tests. Modified ACO can be used to compute with a large number of nodes with optimal and stable computing time results when compared to conventional ACO, this is caused by modifications made to the number of ants so that it is adjusted to the number of nodes that are inputted. But in terms of forming the final route, Conventional ACO still gives better results. The test results in this study found that the average computation time for the last iteration of each test was 0.693 second, and for the 10 iteration computation time obtained an average of 6,977 second for 1 tests. The amount of memory used in 12 tests in this study was 428,969 mb for 10 iterations. Further research is expected when developing the results to form better results. As such, development when further modifying the ACO algorithm should focus on the number of ants, and to test on larger samples.

BIBLIOGRAPHY

[1] M. Dorigo and T. Stützle, *Ant Colony Optimization*. 2004.

[2] I. Brezina Jr. and Z. Čičková, “Solving the Travelling Salesman Problem Using the Ant Colony Optimization,” *Manag. Inf. Syst.*, vol. 6, no. 4, pp. 10–14, 2011.

[3] Y. Feng, “Ant colony for the TSP,” 2010.

[4] D. Mulia, "Aplikasi Algoritma Ant System (AS) Dalam Kasus Travelling Salesman Problem," 2011.

[5] N. N. Poddar and D. Kaur, "Solving the Traveling Salesman Problem using Reinforced Ant Colony Optimization techniques," *World Congr. Comput. Sci. Comput. Eng. Appl. Comput.*, vol. 2013, no. 2013, pp. 33–39, 2013.

[6] A. M. Mohsen, "Annealing Ant Colony Optimization with Mutation Operator for Solving TSP," *Comput. Intell. Neurosci.*, vol. 2016, 2016, doi: 10.1155/2016/8932896.

[7] H. Xu, P. Pu, and F. Duan, "Dynamic Vehicle Routing Problems with Enhanced Ant Colony Optimization," *Discret. Dyn. Nat. Soc.*, vol. 2018, pp. 1–13, 2018, doi: 10.1155/2018/1295485.

[8] M. Yousefikhoshbakht, F. Didehvar, and F. Rahmati, "Modification of the ant colony optimization for solving the multiple traveling salesman problem," *Rom. J. Inf. Sci. Technol.*, vol. 16, no. 1, pp. 65–80, 2013, [Online]. Available: http://www.imt.ro/romjist/Volum16/Number16_1/pdf/05-MYousefikhoshbakht.pdf.